

2/PRTS

10/528988
JC17 Rec'd PCT/PTO 24 MAR 2005

-1-

**METHOD AND DEVICE FOR THE AUTOMATIC GENERATION OF
PROGRAM CODE DOCUMENTATION TEXT AND MANAGEMENT
INFORMATION BASES USING THE DATA OF A DATABASE**

- 5 The invention concerns a method for administration of at least one data object of
an electrographic printer or copier as well as a device to administer at least one
data object. In electrographic printers or copiers, in particular in electrographic
high-capacity printers and high-capacity copiers with a print capacity of greater
than 50 pages DIN A4 per minute, a plurality of data objects are necessary to
10 control the printer or copier. The control units of such printers or copiers contain a
program code that is executed by the respective control unit to control the printer
or copier. Upon creation of the program code, the data objects, in particular the
variables and constants, are added into this or into the source code from which the
program code is generated. In known high-capacity printers, the individual control
15 units and structural components are connected with one another via at least one
printer-internal network. The management and administration of the control units
and network components occurs, for example, with the aid of what is known as the
Simple Network Management Protocol (SNMP).
- 20 What is known as a management information base (MIB) is preferably used for
management and administration. A management information base contains data
sets with whose help a network management system manages all objects to be
managed, for example control units, peripheral devices, servers, hubs, bridges,
routers and further components in the network. These data sets also contain data
25 objects of the printer controller such as, for example, constants and variables.

Furthermore, documentations (such as, for example, service manuals and data
sheets) that likewise contain data objects of the control units of the printer or
copier have been created for known printers or copiers. A comparison of these
30 data was very elaborate, whereby in particular all used variables and constants in
the program modules as well as in the documentation respectively have to be

checked and, if applicable, adapted given changes of program parts as well as given various delivery states of the same high-capacity printer type or, respectively, copier type. The management information base likewise had to be adapted for network management. Errors thereby frequently occur, in particular in that individual data objects have not been correctly changed. The error-prone adjustments of setting parameters of the printer or copier resulting from this have led to faulty printer products as well as to a reduced operating life of individual printer components and consumable materials.

- 10 A print management system for administration of print protocols is known from the English-language translation of the summary of the patent application JP 2002259103 A. The print management system comprises a server that receives a protocol entry from a client given each executed print job as well as data from a databank (management information base) in predetermined time intervals, whereby
- 15 the databank comprises information for network administration of a stored print job or a plurality of stored print jobs of a network printer. The server generates a list with protocol entries from the data of the databank and the protocol entries transferred by the client.
- 20 Furthermore, in the examination proceedings of the parallel German patent application 102 45 527.9 in front of the German Patent and Trademark Office, reference is made to the document "Stewart, B.: Definitions of Managed Objects for Parallel-printer-like Hardware Devices, April 1992 [online], available on the Internet: [URL:http://www.kblabs.com/lab/lib/rfcs/1300/rfc1318.txt.html](http://www.kblabs.com/lab/lib/rfcs/1300/rfc1318.txt.html)",
- 25 researched on 15 April 2003.

It is the object of the invention to specify a method and a device for administration of data objects of an electrographic printer or copier, via which errors in the matching of the data objects are prevented in a simple manner.

This object is achieved via a method with the features of the patent claim 1.

Advantageous developments of the invention are specified in the dependent patent claims.

5 Via a method for administration of data objects of an electrographic printer or copier with the features of the patent claim 1, it is achieved that the data of the data object are changed in the databank and that the changed data can be very simply adopted in the source code via the identifier. Thus the inventive method can, for example, be very simply executed by a data processing system that searches the
10 first source code for the identifier of the data object and replaces the identifier with data of the data object that are stored in the databank, whereby the data processing system automatically generates the second source code. The second source code then contains exactly the data that are currently stored in the databank regarding the data object. It is therewith ensured that, given a change of the data of the data
15 object, a second source code is created in a simple manner in which the data of the first data object are current at each point of the second source code, corresponding to the data state of the databank. In particular variables and constants as well as setting parameters can thereby be very simply administered and, if applicable, changed in the databank.

20 According to a second aspect of the invention, the data objects stored in the databank can be used to generate further source codes. Thus, with the aid of the databank, a plurality of program codes for control units of the printer, the documentation and the general management of the printer as well as a management
25 information base can be generated with exactly the same data objects, whereby errors and inconsistent statements about setting parameters, variables and constants are prevented.

A third aspect of the invention concerns a device to administer data objects of an
30 electrographic printer or copier. The at least the data of one data object are stored in a first storage region. An identifier is associated with the data object. A first

source code that comprises the identifier of the data object is stored in a second storage region. From the first source code, the device generates a second source code in which at least the identifier of the data object are [sic] replaced by the data of the data object.

5

Through this device, given a change of data objects, for example given a change of a variable or a constant, a source code no longer has to be searched through for this data object and every one of these data objects no longer has to be changed, if applicable, multiple times (as in the prior art); rather, only the data object in the databank is changed. A second source code that then contains the updated data of the data object is subsequently repeatedly generated from the same source code. Errors can thus be prevented. In particular when these data objects are contained in multiple source codes, discrepancies in the second source codes and between multiple source codes in which the same data objects are contained are prevented in a simple manner. Elaborate corrections of the source codes are no longer necessary.

10

15

For better understanding of the present invention, reference is made in the following to the preferred exemplary embodiment shown in the drawings, which is described using specific terminology. However, it is noted that the protective scope of the invention should not thereby be limited, since such variations and further modifications to the shown device and/or the method as well as such further applications of the invention as they are shown therein are viewed as typical present or future expert knowledge of a competent average man skilled in the art.

20

25

The Figures show exemplary embodiments of the invention, namely:

Figure 1 a block diagram to generate program modules, a text document and a management information base according to the present invention; and

30

Figure 2 a representation of an output window of a program module to
administer objects with the aid of a network management system.

A block diagram of a system 10 for generation of program modules 22, 24 of a
5 documentation 26 as well as a management information base 28 is [sic] shown in
Figure 1. Data objects are stored in a databank 12. The data objects comprise
variables V1 through V10 and constants K1 through K6. For the data object, in
particular the size, the type, the name, the position within a data object structure,
the write and read accesses, the measuring unit, the limit values, the storage
10 position of the data object in a non-volatile storage, the numerical value, the reset
value, a value table and/or function calls for handling of the data object are
contained in the databank. Respectively one numerical value is associated with
both the variables V1 through V10 and the constants K1 through K6 in Figure 1.
The designation V1 through V10 as well as K1 through K6 serves as an identifier
15 of the respective data object. However, other character strings can also be used as
an identifier in other exemplary embodiments.

With the aid of a first programming environment, a source code 14 of a first
program is created by a programmer for a first control unit with a plurality of
20 instructions AW11 through AW16, for example in a programming language C⁺⁺,
Java or Delphi. Instead of directly inserting the numerical values of the variable
V1 and of the constant K6 into the instruction AW11 of the source code 14, the
identifier of the variable V1 and of the constant K6 is inserted into the instruction
AW11. Instead of the numerical values of the data objects V5 and V2, the
25 instruction AW12 contains the identifiers V5 and V2. The instruction AW13
contains no data object. Instead of the numerical value of the data object K2, the
instruction AW14 contains the identifier K2. The identifier V9 is inserted into the
instruction AW15 instead of the numerical value of the data object V9. The
instruction AW16 of the first source code 14 comprises the identifiers V2 and V1.

A second source code 22 is preferably created from the source code 14 of the first program with the aid of a data processing system. To generate the second source code 22 from the first source code 14, the identifiers of the variables V1 through V10 as well as the identifiers of the constants K1 through K6 are replaced by data, i.e. by the numerical values of the variables that are stored in the databank 12 for the data objects V1 through V10 and K1 through K6. The data processing system accesses the data stored in the databank 12 via a solid body interface. Such an interface is, for example, the ODBC interface.

- 10 Depending on the identifier of the data object, the identifier can be replaced by the complete data set of the data object or by only parts of the data set of the data object V1 through V10; k1 through K6.

- 15 In the exemplary embodiment according to Figure 1, for simple comprehension each data set comprises only the numerical value and the identifier of the data object. The second source code 22 this comprises instructions AW11 through AW16 that replace the numerical values 12 through 26 stored in the databank 12 for these identifiers V1 through V9, K2, K6 instead of replacing the identifiers V1 through V9, K2, K6. The instruction AW11 thus contains the numerical values 12 and 26, the instruction AW12 contains the numerical values 16 and 13, the instruction AW14 contains the numerical value 22, the instruction AW15 contains the numerical value 19, and the instruction AW16 contains the numerical values 13 and 12. The instruction AW13 was already associated with no identifier in the source code 14, such that this instruction AW13 has been associated with no numerical value from the databank 12 in the second source code 22.

- 30 A third source 16 of a second program contains instructions AW21 through AW25. The instruction AW21 contains the data objects K1 and K3, the instruction AW22 contains the data objects V3 and K6; the instruction AW23 contains the data objects V1 and V9 and the instruction AW25 contains the data object V4. In the same manner as has already been described for the first source code 14, a fourth

source code that contains the instructions AW21 through AW24 of the third source code 16 is generated from the third source code 16 of the second program, whereby the identifiers of the data objects V1 through V9, K1 through K6 have been replaced by the numerical values stored in the databank 12 for these identifiers V1 through V9, K1 through K6. The instruction AW21 thereby contains the numerical values 23 and 21, the instruction AW22 contains the numerical values 14 and 26, the instruction AW23 contains the numerical values 12 and 19 and the instruction AW25 contains the numerical value 15. In the third source code 16, the instruction AW24 already contains no identifier, such that the instruction AW24 in the fourth source code 24 contains no numerical value of a data object of the databank 12.

A fifth source code 18 is a text document created with the aid of a text processing program, which text document contains a service and maintenance manual as well as technical data of the printer. The text document contains text blocks Tx1 through Tx6, whereby the text block Tx2 contains identifiers V2 and V1, the text block Tx4 contains the identifier K6, the text block Tx5 contains the identifier V10 and the text block Tx6 contains the identifiers V8 and V3. The text blocks Tx1 and Tx3 contain no identifiers. A sixth source code 26 that contains the text blocks Tx1 through Tx6 is generated from the source code 18 of the text document in the same manner as for the source codes 14 and 16 of the programs 1 and 2, whereby the identifiers V1 through V10, K6 have been replaced by the numerical values stored in the databank 12 for these identifiers. The text block Tx2 of the second source code 26 thus contains the numerical values 13 and 12, the text block Tx4 contains the numerical value 26, the text block Tx5 contains the numerical value 20 and the text block Tx6 contains the numerical values 18 and 14.

A seventh source code 20 contains data of what is known as a management information base (MIB). The management information base contains data sets with whose help a network management system administers (for example with the help of the Simple Network Management Protocol (SNMP)) all objects to be

administered, such as, for example, control units of the printer, servers integrated into the network of the printer, computers, hubs, bridges and routers. Such a management information base is thus always appropriately provided when the control units of the printer are connected with one another over a network,

5 whereby a plurality of networks coupled with one another can be provided in the printer as well as outside of the printer. The management information base is thus an information base or, respectively, a databank for administration of network elements, whereby the control units and software modules as well as data objects are recorded as network elements in the management information base. However,

10 the network elements are also further data processing systems and control units connected with the printer. With the aid of such a management information base, each printer can also be monitored by a remotely arranged maintenance computer when this maintenance computer is connected with the printer, for example over the Internet. The management information base is a map of the most important

15 variables or all variables and other data objects of the printer, whereby the data objects are arranged in the same tree structure as that with which the control units are organized in the printer in which they are stored or by which they are administered.

20 Further information, such as limit values, a physical unit, a specification text of the variables, the size of the variables as well as the type of the variables are contained in the management information base. This information can be represented with the aid of a suitable software that, for example, generates a graphical user interface on a display unit. The current numerical values of the data objects can be read out

25 with the aid of such an interface, and new numerical values can be input that then are transferred to the respective controller and are stored as new numerical values of the variables or of the data object. The interface with which the management information base is displayed serves as what is known as an SNMP sub-agent that accesses the control units of the printer from an SNMP agent provided by a central

30 administration computer. The variables stored and administered in a control unit

can be further sub-divided into function blocks in order to be able to administer the variables with greater ease and more clarity.

Variable definitions in source codes of program modules, for example variable
5 definitions for programming languages such as Delphi or C++, can also be
automatically generated (with the aid of scripts) from the data stored in the
databank 12, which data are associated with the individual variables. Such scripts
can, for example, be generated with a scripting language, such as with the scripting
language Pearl [sic]. Furthermore, items of information for position of the control
10 unit in the network, with which information is associated the respective data object
V1 through V10, K1 through K6, are stored in the databank, whereby the physical
network structure of the control units or, respectively, of the network elements can
be simulated in the management information base. A simple and clear
administration of variables and of the controllers is thus possible with the aid of the
15 management information base. In order to be able to actually implement
maintenance and setting tasks with justifiable expenditure, such a clear
management information base is in particular necessary given a tree-like
arrangement of the controllers.

20 Variables for other printer or copiers can also be contained in the databank 12,
whereby the association with the respective printer (for example via a databank
field) is then contained in the data set of the respective variable. For example, with
the aid of data filters, the data objects V1 through V10, K1 through K6 that should
be adopted into the management information base for the respective printer can
25 then be simply determined for the management information base. An association
of the data objects V1 through V10, K1 through K6 with the respective program
modules 14, 16, 22, 24 can occur in the same manner with the aid of information
that are contained in the data sets of the data objects in the databank 12.

30 A graphical user interface for representation and administration of management
information bases is shown in Figure 2. In the representation of the management

information base according to Figure 2, the network elements and data objects are shown as directories and directory entries in the same tree-like hierarchical structure as that with which the network elements and data objects are arranged in the network of the printer. In region 34, the directory and file tree shown in Figure 2 contains all control units as directories and contains the variables as entries in the directories of the control units.

A first central part 30 of the management information base is predetermined by international standardization of the management information base. A further subdivision in the region 32 is carried out by an internal organization of the firm Océ Printing Systems. A directory for a printer controller management information base with the designation deMib was generated in the sub-directory ops. All further sub-directories and entries in this printer controller management information base have been automatically generated from the information of the databank 12. Thus configuration data objects are contained in the sub-directory deHM configuration. Data for function control of the main module of the printer are contained in the directory deHM-FunctionControl. Current setting values are contained in the sub-directory deHM-Status. Data objects for maintenance and diagnosis of the printer are contained in the sub-directory deHM-Diagnostic. The version number of the management information base is contained in the directory deHM-Version. The variables that are stored in a battery-buffered RAM storage element are contained in the sub-directory deHM-NVRAM. The listed sub-directories contain data objects and/or further sub-directories in which are contained the data objects and/or information regarding the data objects.

Diverse data objects data objects are stored in the directory deHM-OutputBins. The directory deHM-OutputBins contains the sub-directory deHM-OutputBinsTable, which in turn contains the directory deHM-OutputBinsEntry, in which the variables deHM-OutputBinsIndex and deValidOutputBin are stored. Respective properties for both of these data objects can be displayed, which properties are contained in the associated data set in the databank 12 from which

the respective entry in the management information base has been generated. In the other cited directories (both in those further shown and those not further shown), a plurality of data objects are contained that contain the variables and constants as well as further data objects. Individual variables can thereby also
5 contain value tables or refer to functions or procedures of program elements that are executed upon invocation of the variable or from which the current value of this variable is determined via execution of the procedure or of the function.

10 A network address of the printer is set in the field 36 of the user interface shown in Figure 2, for which network address the association of the variables contained in the tree of the management information base is explicit. This network address is an IP address according to the Internet Protocol. The region 34 of the management information base is identical for all printers of the same structural type.

15 An interface in which the management information bases (that should be contained in the management information base directory tree shown in Figure 2) can be selected is displayed (not shown) with the help of a register map that is activated with the aid of a rider 38. A graphical user interface of a program for a communication test with a network participant (for example with a printer
20 connected with the network) is invoked with the rider 40.

Although a preferred exemplary embodiment is displayed and described in detail in the drawings and in the preceding specification, this should be viewed as purely exemplary and not as limiting the invention. It is noted that only the preferred
25 exemplary embodiment is shown and specified, and all variations and modifications that presently and in the future lie within the protective scope of the invention should be protected.

Reference list

| | | |
|----|-------------------|---|
| | 10 | system |
| | 12 | databank |
| 5 | 14, 16, 22, 24 | processing source codes |
| | 18, 26 | documentation texts |
| | 20, 28 | management information base |
| | 30 | predetermined region MIB |
| | 32 | internal structure MIB |
| 10 | 34 | structure MIB printer controller |
| | 36 | network address of the selected printer |
| | 38 | selection of the shown MIB |
| | 40 | selection communication test |
| | AW11 through AW25 | instructions |
| 15 | Tx1 through Tx6 | text elements |